EXHIBIT A

US009323332B2

(12) **United States Patent**
Braun et al.

(10) **Patent No.: US 9,323,332 B2**
(45) **Date of Patent: Apr. 26, 2016**

(54) **FORCE FEEDBACK SYSTEM INCLUDING MULTI-TASKING GRAPHICAL HOST ENVIRONMENT**

(71) Applicant: **IMMERSION CORPORATION**, San Jose, CA (US)

(72) Inventors: **Adam C. Braun**, Portland, OR (US); **Jonathan L. Beamer**, Lyndhurst, OH (US); **Dean C. Chang**, Gaithersburg, MD (US)

(73) Assignee: **Immersion Corporation**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/589,355**

(22) Filed: **Jan. 5, 2015**

(65) **Prior Publication Data**

US 2016/0004310 A1 Jan. 7, 2016

Related U.S. Application Data

(63) Continuation of application No. 13/225,899, filed on Sep. 6, 2011, now Pat. No. 8,928,581, which is a continuation of application No. 11/824,593, filed on Jun. 28, 2007, now Pat. No. 8,020,095, which is a

(Continued)

(51) **Int. Cl.**
G09G 5/00 (2006.01)
G06F 3/01 (2006.01)

(Continued)

(52) **U.S. Cl.**
CPC ***G06F 3/016*** (2013.01); ***G06F 3/03543*** (2013.01); ***G06F 3/0481*** (2013.01); *G06F 2203/014* (2013.01)

(58) **Field of Classification Search**
CPC G06F 3/016
USPC 345/156
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,334,280	A	6/1982	McDonald
5,157,384	A	10/1992	Greanias et al.

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 92/00559	1/1992
WO	WO 95/32459	11/1995

(Continued)

OTHER PUBLICATIONS

File History for U.S. Appl. No. 60/065,122, filed Nov. 12, 1997, in the name of Adam W. Divebliss, et al.

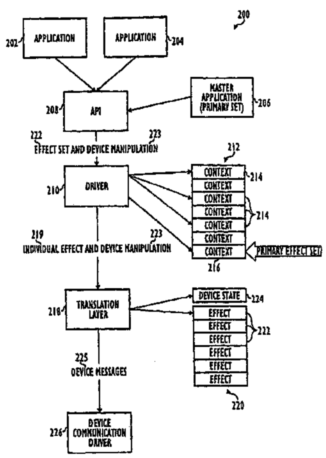
(Continued)

Primary Examiner — Jason Olson
(74) *Attorney, Agent, or Firm* — Miles & Stockbridge P.C.

(57) **ABSTRACT**

A force feedback system provides components for use in a force feedback system including a host computer and a force feedback interface device. An architecture for a host computer allows multi-tasking application programs to interface with the force feedback device without conflicts. One embodiment of a force feedback device provides both relative position reporting and absolute position reporting to allow great flexibility. A different device embodiment provides relative position reporting device allowing maximum compatibility with existing software. Information such as ballistic parameters and screen size sent from the host to the force feedback device allow accurate mouse positions and graphical object positions to be determined in the force feedback environment. Force feedback effects and structures are further described, such as events and enclosures.

11 Claims, 12 Drawing Sheets



A force feedback feature that can be implemented in either or both of the host and the force feedback device is "clipping." Clipping is the selective omission of reported position data when displaying the graphical object or other user-controlled graphical object to cause a sensory effect on the user in certain circumstances. Such circumstances include those where displaying the graphical object in correlation with the precise user object position would not provide a desired effect. The most common case for clipping is when the graphical object encounters a surface or object in the graphical environment which is desired for the user to experience as a hard obstruction or wall. The wall is desired to prohibit further movement into the surface or object; however, due to limitations in hardware, a strong enough force cannot be output on the user object to actually prevent motion into the wall.

Clipping can be easily performed by the force feedback device to ease computational burden on the host. For example, the local microprocessor **130** can check whether the user object is being moved into a wall in the graphical environment; if so, the microprocessor can simply discard the position data and report a constant position (or delta) to the host computer that would cause the host to display the graphical object against the wall surface. This allows the host computer to not be involved with the clipping process and simply display the graphical object at the reported position, thus freeing the host to perform other tasks.

The host computer **18** can also perform clipping in an embodiment. Since it is desirable to keep the application programs **202** and **204**, the operating system, and other high level programs ignorant of the clipping process, a lower level program preferably handles the clipping. For example, the translation layer **218** as shown in FIG. **4** (or alternatively the context driver or API) can check for the conditions that cause clipping to be applied. If clipping is to be performed, the translation layer alters the input position data appropriately before it is sent on to the context driver, API, operating system and any application programs. Problems with this implementation include increased load on the host with the overhead of intercepting and translating incoming messages.

A different force feedback feature that can be implemented either by the force feedback device or the host computer is "pressure clicks" or "click surfaces." As described above, these are surfaces, objects or regions in a graphical environment which have additional functionality based on the position of the graphical object in relation to the object. Thus, a border of a window can be designated a click surface such that if the user overcomes a resistive force and moves the graphical object (or user object) over a threshold distance into the border, a function is activated. The function can be scrolling a document in the window, closing the window, expanding the window size, etc. A similar click surface can be used on an icon or button to select or activate the icon or button.

The force feedback device can implement click surfaces to ease computational burden on the host computer. The local microprocessor can check whether the graphical object has moved into a click surface or region and output the resistive force as appropriate. When the graphical object is moved past the threshold distance, the microprocessor reports to the host computer that the action associated with the click surface has been made; for example, the microprocessor reports that a button press or double click has been performed by the user. The host receives a signal that a button has been pressed and acts accordingly. Or, the microprocessor reports that a particular click surface has been specifically selected, where the host can interpret the selection and implement an associated function.

The host may also implement at least a portion of the functionality of click surfaces. The microprocessor can output the resistive (or other type) force associated with the click surface, but only reports the user object position to the host. When the host determines that the graphical object (or user object) has moved beyond the threshold distance, the host implements the function associated with the click surface (scrolling text, closing a window, etc.) As with clipping, it is preferred that the translation layer handle this functionality by modifying data accordingly before passing it to the context driver and API.

While the subject has been described in terms of embodiments, it is contemplated that alterations, permutations and equivalents thereof will become apparent to those skilled in the art upon a reading of the specification and study of the drawings. For example, although examples in a GUI are described, the embodiments herein are also very well suited for other two-dimensional graphical environments and especially three-dimensional graphical environments, where a user would like fine positioning in manipulating 3-D objects and moving in a 3-D space. For example, the isometric limits are quite helpful to move a graphical object or controlled object in a 3-D environment further than physical limits of the interface device allow. In addition, many different types of forces can be applied to the user object **12** in accordance with different graphical objects or regions appearing on the computer's display screen and which may be mouse-based force sensations or graphical object-based force sensations. As stated above, many types of user objects and mechanisms can be provided to transmit the forces to the user, such as a mouse, touch screen or other touch device, trackball, joystick, stylus, etc. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and are not limiting.

What is claimed is:

1. A software method in a multi-tasking environment comprising:

concurrently running a plurality of application programs, wherein each application program includes one or more data sets, each data set comprising a representation of one or more force effects;

receiving from an active application program a force effect command;

generating a signal representing the force effect command; and

outputting a force effect based on the signal.

2. The software method of claim **1**, wherein the multi-tasking environment includes a graphical user interface.

3. The software method of claim **1**, wherein the force effect command is an instruction.

4. The software method of claim **1**, wherein the force effect command comprises one or more force effect parameters.

5. The software method of claim **1**, wherein the force effect is a single force or a series of forces.

6. The software method of claim **1**, further comprising storing the plurality of applications programs in memory.

7. The software method of claim **1**, further comprising

making one of the plurality of concurrently running application programs the active application program.

8. A software method in a multi-tasking environment comprising:

storing a plurality of representations of force effects in memory, wherein each representation of a force effect is associated with an application program;

running a plurality of application programs in the multi-tasking environment concurrently;

making one of the plurality of concurrently running application programs active;

receiving from the active application program a force effect command; and

generating a signal representing the force effect command.

9. The software method of claim **8**, further comprising:

receiving from an inactive application program a second force effect command; and

generating a signal representing the second force effect command.

10. A haptic computer system comprising:

a computer memory for storing a plurality of application programs;

a means for concurrently running more than one of the a plurality of application programs in a multi-tasking environment;

a means for making one of the plurality of concurrently running application programs the active application program; and

a force sensation generator configured to output one or more force sensations associated with the active application program.

11. A haptic computer system comprising:

a computer memory configured to store a plurality of application programs that command force sensations;

a processor configured to run more than one of the plurality of application programs concurrently;

an operating system configured to provide a multi-tasking environment for the plurality of application programs, wherein one of the plurality of concurrently running applications is an active application program; and

an actuator configured to output one or more force sensations commanded by the active application program.

* * * * *